Intelligent Storage Management Using Cloud Shared Disk Architecture

Subashis Biswas, Nilanjana Roy Chowdhury, Argha Roy and A.B. Bhattacharya

Abstract In recent years, there is tremendous demand of cutting-edge cloud-based applications in many of the industries. We have proposed in the paper a shared disk cloud database architecture as the basis on which an intelligent data storage management system can be developed for enriching cloud-based web applications. Important features of this proposed architecture are single copied data consistency, dynamic load balancing and high benchmark performance. Based on the software layer, an intelligent data management system for popularizing the concept of SaaS has been pointed out suggesting a cost-effective solution for popularizing the cloud environment.

Keywords Storage area network · Shared disk · Storage architecture · Cloud system · Intelligent data management

1 Introduction

The subject of cloud computing refers to the use of web-based applications and/or server services which are paid for accessing, instead of software or hardware, which are bought and installed locally. Today, organizations of all sizes and bases are

Subashis Biswas
Department of CSE, Netaji Subhash Engineering College, Kolkata, India
e-mail: subashiscse@gmail.com

N.R. Chowdhury · Argha Roy
Department of CSE, Techno India University, Kolkata, India
e-mail: nilanjana.rns@gmail.com

Argha Roy
e-mail: arghacse@gmail.com

A.B. Bhattacharya (✉)
Department of Electronics and Communication Engineering, Techno India University,
Kolkata 700091, West Bengal, India
e-mail: bhattacharyaasitbaran@gmail.com

J.K. Mandal et al. (eds.), *Proceedings of the First International Conference on Intelligent Computing and Communication*, Advances in Intelligent Systems and Computing 458, DOI 10.1007/978-981-10-2035-3_9

going to implement this technology. The technology is cost-efficient as it offers unlimited data storage capacity, efficient backup and recovery, besides quick and easy access to any type of information from anywhere and anytime and facilitates fast deployment of web applications [1, 2]. Now-a-days, updated and current database MySQL, DB2, Oracle has robust optimization technology that helps to enable efficiency. In the internet based workload type of environment, sometimes data is hosted on a centralized database as it becomes difficult for an application to partition the data and this is why the scalability of the particular application gets limited. In this paper, we have presented ways to ensure scalability of such applications. In this architecture any number of database nodes can be processed any portion of data. Our approach is based on the data mining technique. Developed algorithm classifies the cloud services and ranks them accordingly [3, 4].

2 Technology Preliminaries

Two types of database management systems are of great significance in present day database market, viz., system of shared nothing and system of data sharing. The first type of the system is comparatively simpler to develop in comparison to the system of data sharing [4]. Data sharing systems are more advantageous than shared nothing systems with respect to the load balancing. First, we consider the Shared Nothing DBMS. The shared disk architecture is shown in Fig. 1 using a block diagram.

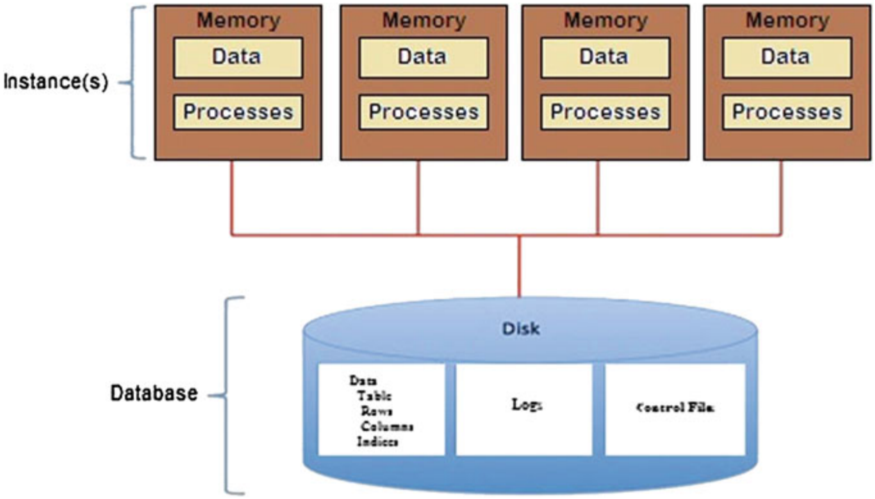


Fig. 1 Block diagram of shared disk architecture

3 Proposed System

In the system every instant should be connected with the processor and the memory [5, 6]. The proposed architecture is shown in Fig. 2. According to our proposed system based on shared disk cloud database architecture, inside a cloud, we have added a new layer which is the software layer. This area has been created in between the memory and the storage. This entire system will function inside a storage area network (SAN). This software layer is connected to the memory and the storage area which consists of all the disks (or specific databases/files). The software layer will maintain the log of data storage and data access including authentication and authorization feature of this software layer. If any particular processor of any other cloud tries to access any specific data of this storage area, then this software layer first checks whether this data access request is authenticated or not. If the data access request is authenticated and authorized, the particular processor is able to access the requested data from the specific database. The location of this requested data will be shown by this software layer. If the data access request is not authenticated then the particular processor will not be able to access the requested data. The data access will be prevented by the software layer. Without the existence of such a software layer, any processor from any other cloud outside the storage area network can access any data it needs from the storage area since there is no question of authorization and authentication.

Outside of SAN area, there will be another opportunity to build low cost cloud environment to popularize the concept. This is because, through this concept, it will

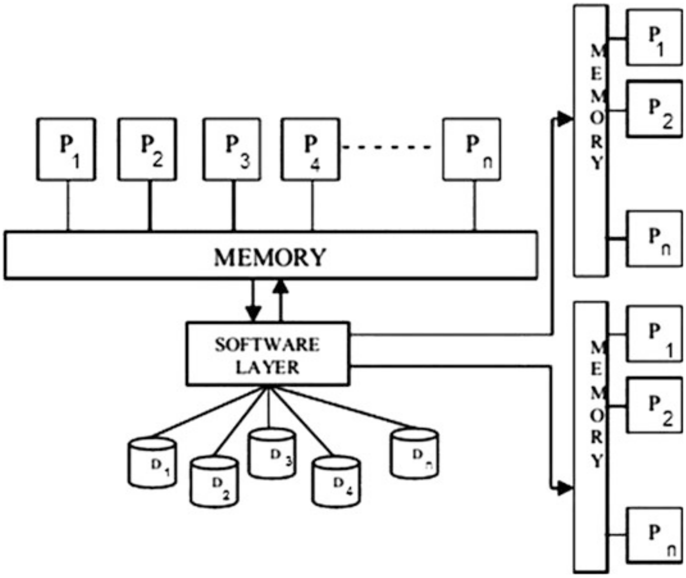


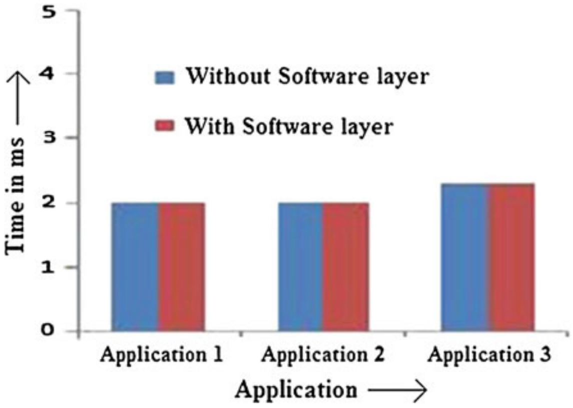
Fig. 2 Proposed architecture

be possible to develop cloud platform without storage. According to this concept, the same storage can be shared by several cloud servers (consisting of memory and processors). Outside the SAN, we need a request for authentication/authorization to secure the transaction. Here, cloud can share common infrastructure which is under the concept of infrastructure as a service (IaaS).

We have deployed the applications over a cloud platform with LAMP facilities. Open source database has been used for testing purpose. The application has been prepared and maintained in a cloud environment and storage has been maintained at other locations through static IP for data processing from various applications. We have tested the application in a real world cloud environment. Proposed layer and performance testing has been done through simulation. According to our simulation, it is giving the same access time. Traditional shared disk architecture are maintaining the same log and it has been processed into the same disk but according to our new layer, unauthentic hit will not be done into the disk. It will reduce to congestion too. According to the simulation graph, it is showing that into the SAN, frequently access database application is structuring data without high latency. SAN Application through software layer is presented in Fig. 3. During the time of outside access/request, it varies over bandwidth to hit the white list table. But, high bandwidth is getting positive response to access the data disk in comparison with SAN application. But it is possible to implement new proposed architecture which is going to reduce hardware cost and cloud can facilitate IaaS for another cloud.

We have developed this software layer in order to prevent unauthorized access of data from this storage area. If this concept is implemented, then all the data will be stored and maintained in only one cloud and several other clouds containing only processors and memory can be developed at very low cost. Implementation of this concept will ensure access of data by any authorized and authenticated processor in any cloud from a common cloud through the software layer. Access latency time is shown in Fig. 4, where there is a plot of bandwidth versus time for SAN request and outside request.

Fig. 3 SAN Application through software layer



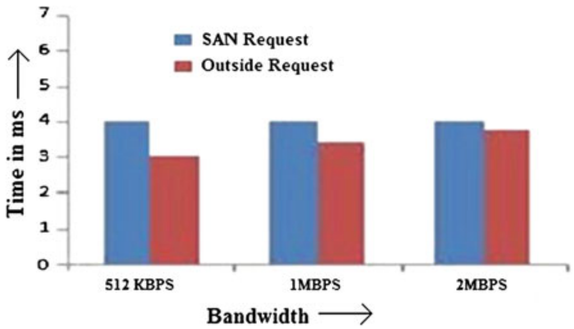


Fig. 4 Access latency time

The associated algorithm of SAN operation is shown below:

Algorithm:

SAN_operation

```
If(request)
    {$flag=0;$t=systemtime;
    $s=”select * from data_log where file=request and
    active=1 ”;
    $rec=mysql_query($s,$con);
    While($row=mysql_query_fetch_array($rec)){
    $a1=row[id]; $a2=row[location];
    $a3=row[signature]; $a4=row[diskid];
    $flag ++;} If($flag==1){
    $p=insert into in_user_log
    values(“”,$t,$a2,$a4,request,$a3)”;
    $re=mysql_query($p,$com);
    If(re){Connect to storage device;}}}
```

Outside_req_operation

```
{If(request) {
    $fg=”phigical”;
    $mycom=ob_get_contents();
    $ip=p_server[‘Remote_add’];
    $mac=$_strpos($mycom, $fg);
    $activate=0;
    $s=”select * from white_list where whiteip=$ ip and
    mac=$mac”;
    $rec=mysql_query($s,$con);
    While ($row=mysql_fetch_arrey($rec))
    {$ a1=row[id];
    a2=row[networkid]; a3=row[accessmode];
    $active=1;} If($activete==1){
    $t=systemtime();
    $tid=genaratetid();
    $p=”insert into tid_genarator values(“ “,$tid,
    $t,$a1)”;
    $re=mysql_arrey($p,$com);}
    If(disconnect)
    {$p=”update tid_genarate set active = 0 where
    $tid=$tid”;
    $re=mysql_query($p,$com);
    fflush($a1);
    fflush($a2);
    fflush($tid);} }
```

4 Analyses

Cloud Computing technology has revolutionized the way in which software can be used to accomplish storage and retrieval of essential information. Cloud computing in its form of SaaS has earned worldwide acceptance and popularity on account of its customer convenience. The cloud offers infinite amount of data storage and this advantage has enabled it to achieve superiority and preference over traditional forms of software. The strength of database architecture is proportional to the scalability and performance of any cloud-based software. In the shared nothing database architecture the cloud-based system has its own private memory in one or more disks. According to this system, the clustered processor performs communication by sending message through a network like interconnected computers. Shared-nothing multiprocessor can scale up to thousand processors since the processors do not interfere among themselves but for each cluster, if separate private memory and disk exist, the system overall becomes expensive to use. According to the Shared Disk architecture, the same disk can be used for all the processors or nodes. Several data tables, logs and control files are maintained on the disk. According to our new concept, other cloud service providers and individual clients can share the disk of other cloud platforms but in this way, other cloud service providers can utilize the same disk. For data access, authentication and authorization checking creates excessive pressure on the disk which can result into data disk crashing and hacking. Outside the disk, a software layer is proposed. When outside users of the same SAN access the disk for retrieving data, then data processing will be done inside the software layer. Log files, table information, metadata and location will be maintained inside this software layer. If any data request comes from outside IP without authorization, then it will get rejected at this software layer and the disk access will be prevented. If any communication needs to take place inside the SAN network, then there will not be any need for authentication and authorization. Here, two algorithms are proposed: the SAN operation and the outside req operation. All the data processing will be accomplished inside the SAN network without the need of authentication and authorization. In this case, a hit to a specific disk is possible from only disk location and metadata and this improves the processing time. We have developed and tested our application in a real cloud environment with Xen architecture consisting of 4 hexacore CPUs of 4 GB RAM each on 1 TB storage platform. We have used MySQL database in the communication layer and PhP with Ajax and HTML in the front end for developing our application. We have developed our proposed software layer in this cloud architecture and observed the functioning of our application through this software layer and concluded that there is problem with data access time and that there is uniformity in data layer access time. We have converted various access times into 100 scale with or without the software layer when we find that our application experiences the same data access time. We have also analyzed its corresponding graphical representation. But, time variation was obtained when any data request from outside communicates through a separate algorithm. In this case, the storage is accessed from a web server, from a different IP, through static IP and noted that the time variation completely depends on the bandwidth. In comparison to SAN request, latency time is greater. In case of high bandwidth, the latency time of outside requests can be reduced but it is optimum as per SAN request.

We have used a simulator based on our proposed algorithms. Based on the statement, we have determined conditional code access time [6]. The operation time of outside req operation algorithm is greater than that of SAN operation. By calculating this operation time, we have identified that the latency time is greater. By implementing the concept of Infrastructure-as-a-Service, various clouds will be able to share the storage and it will be possible to develop other low cost cloud environments. Through the software layer, specific data requests are able to hit specific locations. An unauthorized filtration system is maintained to prevent direct hits to the data disk. This reduces power consumption. It may be pointed out that cloud computing related to the cutting-edge cloud-based applications in industries may be regarded as one of the most booming technology and plays a vital role in the possible regulatory changes and thus implementing better applications by using the potential of cloud computing [7].

5 Conclusions

The functioning of shared-disk database architecture depends on inter-nodal messages. These messages alert a node about the status of all the other nodes. In the cluster the number of nodes is found to be proportional to the time taken for a message to reach a particular node from another particular node. As a result, long wait-states are created. This feature affects the scalability of shared-disk databases. On this shared disk architecture, we have worked on a file system. We can bind a flag with this file system so that after getting authentication, it can directly access the data storage. This is because authentication signature will be bound to the file system and several research works will be possible for creating this file system.

Acknowledgments We like to express our sincere thank to Mr. Mayur Goutam, System Engineer, CTRLS and Mr. Avik Chakraborty, System Engineer, CTS for their guidance and help in many parts of the work during implementation Phase.

References

1. Arora, I., Gupta, A.: Cloud Databases: A paradigm shift in databases. International J. of Comp. Sc. 9, 1694-0814 (2012).
2. Melnyk, R. B., Zikopoulos, P. C.: DB2: The Complete Reference. McGraw-Hill, (2001).
3. Microsoft SQL Server, Microsoft, http://www.microsoft.com.
4. Haghighat, M., Zonouz, S., Abdel-Mottaleb, M.: Cloud ID: Trustworthy Cloud-based and Cross-Enterprise Biometric Identification. Exp. Sys. with App. 42, 7905–7916 (2015).
5. Schmidt, E., Rosenberg, J.: How Google Works, Grand Central Publishing, 11 (2014).
6. Franck, C., Song, H., Ferry, N., Fleurey, F.: Evaluating robustness of cloud-based systems. J. of Clou. Comp. 4, doi:10.1186/s13677-015-0043-7 (2015).
7. Ward, J. S., Barker, A.: Observing the clouds: a survey and taxonomy of cloud monitoring. J. of Clou. Comp. 3, 1–30 (2014).